LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-224673

Mobile Data Collection Applications: A Proof of Concept

J. Chang

September 24, 2006

REACHS
Livermore, CA, United States
October 2, 2006 through October 3, 2006

Updated October 14, 2003

Mobile Data Collection Applications: A Proof of Concept

Jonathan Chang
University of California, Berkeley
Berkeley, CA 94720

T. Lowe
Hazards Control Department
Lawrence Livermore National Laboratory
Livermore, CA 94550

September 19, 2006

Abstract

Mobile Data Collection Applications: A Proof of Concept. JONATHAN CHANG (University of California, Berkeley, CA, 94720) TIM LOWE (Hazards Control Department, Lawrence Livermore National Laboratory, Livermore, CA 94550)

This project's goal is to provide a proof of concept for mobile data collection applications, and identify the best ways such applications could be implemented and used. Such an application should decrease the time and resources users now need to devote to redundant data processes, and provide an easy of locating and retrieving data at a later time. The two types of available mobile devices, Personal Digital Assistants and Tablet Personal Computers, each have their particular strengths that suggest themselves for certain types of applications. As such, parallel data collection applications have been developed, with a common web application for uploading information to the database. While these aspects have been developed and proven, it still remains to refine these applications, develop the tables to hold their data, and field-test with users for their feedback.

Project Description

As a prominent research facility, Lawrence Livermore National Laboratory (LLNL) does ground-breaking work in several scientific fields on a daily basis. With work like this, however, comes the need for a comprehensive safety architecture to support the lab's novel, but potentially hazardous research. To achieve this, the Hazards Control Department performs innumerable regular audits, surveys, and other tests to monitor conditions onsite. Unfortunately, many of these tests are still performed on paper, then filed away inconsistently, making it difficult if not impossible to reconcile these results at a later date. In the case of laser safety audits, the primary focus of this project, the approximately 300 original hand-written audits are manually entered into a Filemaker Pro database maintained for their group. However, this does not solve the problem of consistency or eliminate the redundancy of hand-written reports.

This project's goal is to develop a mobile data collection application, originally aimed at the laser safety audits performed by LLNL's Hazards Control department but with the potential for other applications. In order to cover as much potential application as possible, the project develops parallel solutions for both a Tablet Personal Computer, the HP Compaq tc1100 running Windows Tablet Edition, and a Personal Digital Assistant (PDA), the Symbol PPT 8800 running Windows Mobile 2003. The resulting applications are divided into three parts by functionality; data collection, data upload, and data retrieval. During collection, data in the field is recorded into a mobile device and stored in some format on the device. Afterwards, the device is returned to a connected work area and the data uploaded and persisted onto the appropriate database. The final part requires some interface for searching and retrieving previously recorded entries, as well as providing printed reports in case hard copies are needed. While developing these applications, it was important to take into account the move Information & Communication Services (ICS) is currently making towards Java-based applications; ideally, any new application should be developed using Java, JavaServer Faces (JSFs), or related technologies. Beyond this, each of these functions also has its own design considerations that were taken into account when determining how each would be implemented.

Mobile Device Comparison

While this project develops applications for both PDAs and Tablet PCs, it is still worthwhile to compare the two technologies and design applications that will take the best advantage of the individual hardware. Besides the relevant technical differences summarized in Fig. 1, there seem to be three main areas of contention suggested by research of the two platforms and their current applications: portability, utility, and hardware features.

	Symbol PPT 8800 PocketPC	HP TC1100 Tablet PC
Weight	10.8 oz. (16 oz./lb.)	4 lb.
Dimensions	1.3 in. H x 3.1 in. W x 5.7 in. L	0.8 in. H x 10.8 in. W x 8.5 in. L
Screen size	3.5 in.	10.4 in.
Input	Screen	Screen, keyboard
Memory	64 MB total	1 GB RAM, 40 GB HD
Features	Barcode scanner, IR port	Attachable USB scanner

Fig. 1 Technical comparison between the PDA and Tablet PC used in this project.

Since users will probably be carrying and using these devices for extended periods of time, portability is an important issue. Most tablet PCs run around three or four pounds, with examples as light as a pound and a half and as heavy as seven. At a preliminary meeting of intended users, some concern was expressed regard the effect of this weight over the course of a day in the field. PDAs, on the other hand, are weighed in terms of ounces and grams, never more than a pound. Additionally, the battery life of a Tablet PC is far less than that of a PDA; the tc1100's battery life is advertised at around 4.5 hours, perhaps less with heavy usage, while it is nearly impossible for a PDA to run down its battery over the course of a day, especially with wireless and Bluetooth options disabled.

Regarding utility, PDA screen run between 3 and 4 inches diagonally across, while tablets run up to 14 inches diagonally, and even larger for widescreen options. With such limitations, a PDA could never be used with forms containing more than a few fields of information. Entering comments would also be much better suited to the larger writing area, and even keyboard, available on the tablet. However, it is possible that a PDA with a keyboard, like the Symbol MC 9000, may offset this consideration, and user opinions will be collected when a number of working demos are ready.

In terms of hardware, a PDA will have a much stricter constraint on available space than a tablet; in the Symbol PPT 8800 purchased for this project, there is a total of 64 MB of memory on the device, to be allocated between Storage and Programs. As shipped, about 15 MB of that memory is already in use with the operating system and other sundries. A tablet on the other hand, has all the memory available to a conventional laptop or PC. The Symbol PPT 8800, however, includes a barcode scanner and IR port, with associated software for their usage. It is much less likely to find such a feature integrated into a tablet, though USB-connected barcode scanners are available.

Data Collection

Data collection can go in one of two directions: a completely custom-built application, or a pre-existing software package. An application designed specifically for this project has the

advantage of being customized to perform exactly as required, as well as take advantage of the existing information framework. Additionally, a Java application designed to run on a PDA would utilize the J2ME platform, a subset of the standard platforms that run on conventional workstations, laptops, and Tablet PCs. As a result, any application designed to run on the PDA could also work on a Tablet PC, with minor modifications for screen size and the Tablet's trademark "ink" capability. A software package, on the other hand, trades customized functionality for vastly reduced requirements for development time and resources. Developing the requisite application would require significant effort from experienced Java programmers to write, thoroughly test, and finally support. Unfortunately, at this stage in the transition to Java technologies, this group does not possess the human resources to perform such a task, and hiring outside programmers to assist in development would cost significantly more than a decent software package. It seems best, therefore, to pursue a pre-developed solution while keeping in mind the more attractive option of a custom Java application for future development.

Research into the software options available to Tablets and PDAs yielded several options for both platforms. The most likely candidates were then compared according to compatible platforms, relevant data and connection features, technical requirements, and pricing (see Fig. 2). Advice was also sought out from other groups onsite that had developed or were developing similar applications, and in the end this was the most heavily weighted factor. Oscar Nazario was an indispensable resource, having already developed a similar application for Environment, Safety and Health (ES&H) Assessments deployed on Pocket PCs. His application consists of a small application on the PDA created by a kind of Integrated Development Environment (IDE) put out by MCL Software. In the field, the PDA collects data into a text file that it sends to the operator's terminal when it is replaced in its cradle. The text file is then uploaded via a web application to their server, and parsed into a series of queries that inserts the data into ES&H's database. This plan was particularly attractive because Mr. Nazario had already proven it worked, and so it became the guide for our own application development.

Despite further comparisons, Mr. Nazario's choice of MCL as an initial software package remained the best choice, as an ideal compromise between end-to-end development and a complete software package. Using MCL-Designer, a user with little programming background

Fig. 3a Screens of the PDA application.

Company	Product	Device	Platform	Data Features	Database access	Synchronization	Space Requirements	Pricing
Pendragon	Forms	PDA	Palm/PocketPC	Makes its own Access database on PC, or synchronizes with external Access database or other ODBC data source. Max 96 forms/device, max 250 fields/form	Links database-to-database; central management allows change in one place to update devices on the next synchronization.	Pendragon SyncServer acts as a network server on port 201, listening to direct network connections. More advanced versions of SyncServer allow concurrent connections.	**Palm**: 350 KB for application, 64 bytes/field, max 64 KB/record. **PocketPC**: 64 MB RAM	Forms 5.0 Kit: $249 ($229 w/o CD) Add. Licenses: $60/license (2-9) SyncServer: $695
DDH Software	HanDBase	PDA	Windows/Mac (separate programs)	Max 200 databases, max 100 fields	Export to Access or ODBC on Windows, and FileMaker Pro on Mac.		**Palm**: 470 KB for application **PocketPC**: 700-750 KB	Professional: $39.99/lic (1-9) Enterprise: $99.99/lic (1-9)
FileMaker	Mobile 8	PDA	Windows/Mac (Palm only)			Syncs with FileMaker Pro	**Palm**: 2 MB storage **PocketPC**: 16 MB RAM	7 already site-licensed Upgrade to 8: $19
	Visual CE	PDA	Windows	Fully relational database				
MCL	Collection Suite	PDA	PocketPC	Application development environment, can create custom fields and make use of the barcode scanner for input.	Capable of connecting with ODBC, or outputting a text file to be inserted into the database via web-based application (ES&H Working Group)		Depends on designed application.	MCL-Collection Symbol PocketPC/CE: $1,695
PTS	TracerPlus	PDA	Windows	Max four fields/record.	Converts data to CSV text file, available for transfer.	ODBC Link for TracerPlus allows TracerPlus data to link to Access, SQL Server, Oracle, etc.		TracerPlus (PalmOS): $69 TracerPlus (PocketPC): $125 ODBC Link: $69
Active Ink Software	Enterprise Designer, Client	Tablet	Windows				NA	Enterprise Client: $249 Enterprise Designer: $499
Mi-Co	Mi-Forms	Tablet	Windows	Cross-field validation or "business rules" for data-checking	Export as CSV or Excel files, or via ODBC	Two-way communication allowed between form application and back end database.	NA	Designer: $4000 Client: $5000/10 pack
FileMaker	Pro 8	Tablet					NA	Already site-licensed
Design Universe	E-Pen & Forms	Tablet	Windows	Scan in paper forms and build electronic form based on the resulting framework. Builder also allows custom creation of forms.	Export to MS Office programs (Excel probably best option). Professional editions of Builder and Filler include ODBC access, and export to CSV, database, and XML.	Synchronize with Excel table or XML forms.	NA	Builder Consumer: $2999 Builder Professional: Request Filler Consumer: $399 Filler Professional: $599

Fig. 2 Comparison of software solutions for data collection function. Some categories were unavailable or not applicable to the corresponding software.

can design screens, establish target files, and write simple instructions to be carried out on entering or exiting any of the screens. Experimentation with a demo version of Designer confirmed that the degree of customization was up to the requirements of the application, so the entire collection of Designer, Client, and Link were chosen to develop the PDA data collection application, deploy it on the PDA, and handle the transfer of data and application updates, respectively (see Fig. 3a).

The choices for Tablet PC were similar in functionality; each allow for the creation of a form with varying degrees of collection-time validation, and a variety of data storage and persistence methods. Since a plan of file upload through web application had already been decided on, purely database applications like Filemaker Pro and Microsoft Access were eliminated in favor of those that featured more attractive form elements and data file output. While Mi-Forms from Mi-Co Software had the cleanest design and was the most professional-looking product, it also exhibited some performance issues on a well-configured Tablet PC and was much more expensive than other options. The final choice for the tablet application, E-Pen & Forms from Design Universe, was almost as functionally complete as Mi-Forms, did not exhibit the same performance glitches, and was much more cost-effective by comparison (see Fig. 3b). Additionally, after some initial trouble getting in contact, Design Universe showed an impressive willingness to assist in our development process, which was also an attractive selling point. Since then, E-Pen & Forms has served well in developing the Laser Safety Audit, and Design Universe representatives have continue to prove helpful and attentive in addressing any implementation questions.

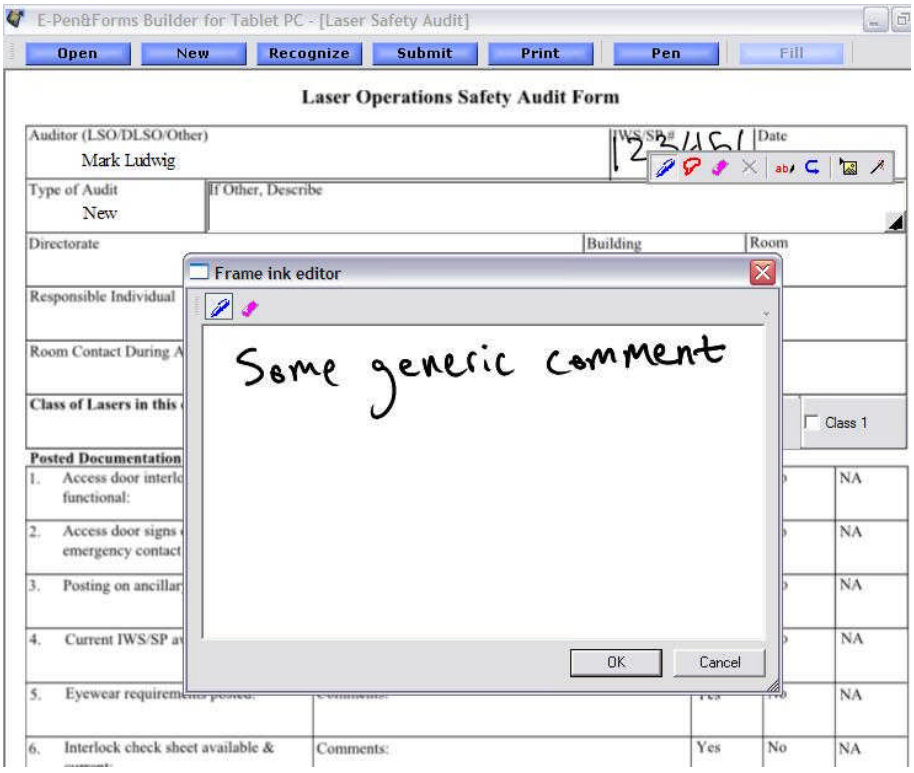

Fig. 3b Screen of the tablet application

At this point, the data collection function of these applications was essentially complete, barring certain details of output type, file names, and other logistics that are dependent upon the application's other functions.

Upload and Database Persistence

In keeping with the departmental conversion to Java-based applications, the upload web application for this project was designed using the Oracle JDeveloper 10.1.3 IDE. Because of its power features and integration with JDeveloper, the relatively new JSF application framework was chosen for development. Among other benefits, the JSF framework implements a clearly delineated structure separating business logic and user interface (UI), and provides a rich, component-based validation framework for user input. The discrete framework allows programmers with different specialties to work on separate parts of the application, depending on their expertise, then synthesize a complete application from their combined work. At the same time, JSF's powerful validation features will be very useful in checking for a variety of data specifications before committing anything to the database. This will both encourage data integrity and provide immediate feedback to users about the status of their data transfer, and what if any actions need to be taken in case of problems.

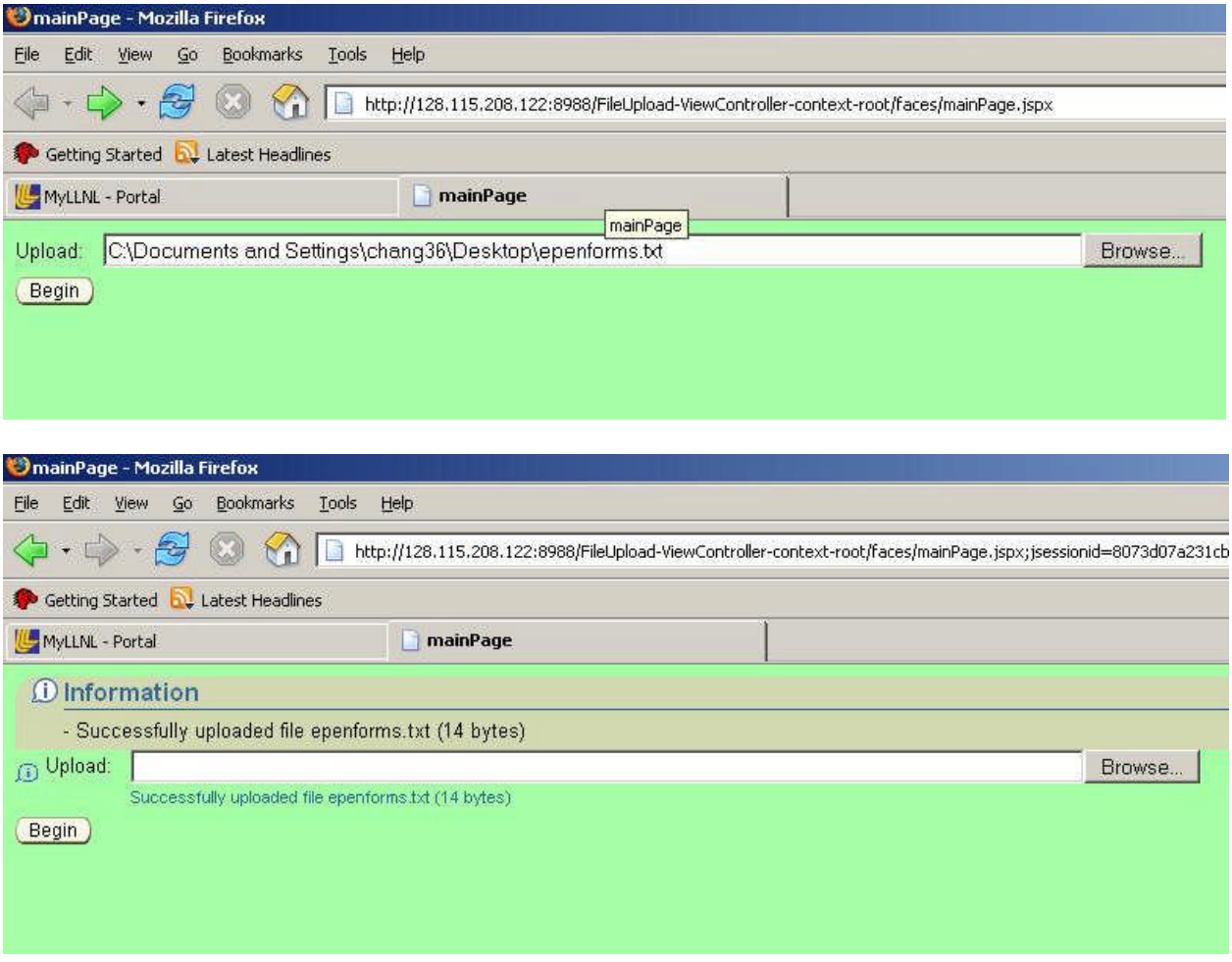

Fig. 4 Example of user uploading a file.

A preliminary proof of the process has been implemented by Jim Collins with functional file transfer and insertion of data into the database (see Fig. 4). The success message, file upload field, and submit button are ADF Faces components in the main JSP, created by the code below (see Fig. 5). To process the file, the upload field is linked to UploadBackingBean.uploadedFile, a method of the Java class UploadBackingBean, as its ValueChangeListener. This

```
<af:inputFile label="Upload:" columns="100"
              valueChangeListener="#{UploadBackingBean.fileUploaded}"></af:inputFile>
<af:commandButton text="Begin" />
```

Fig. 5 ADF user interface components (above) and Java backing method for upload components (below).

```
public void fileUploaded(ValueChangeEvent event)
  {
    String fileDestination = "C:\\tmp\\Uploaded_";
    UploadedFile file = (UploadedFile) event.getNewValue();
    if (file != null)
    {
      FacesContext context = FacesContext.getCurrentInstance();
      FacesMessage message = new FacesMessage("Successfully uploaded file " +
                                              file.getFilename() + " (" +
                                              file.getLength() + " bytes)");
      context.addMessage(event.getComponent().getClientId(context), message);
      try {
        // create input stream reader object
        InputStreamReader isr = null;
        // get input stream from UploadedFile object
        InputStream fileIs = file.getInputStream();
        // read the input stream
        isr = new InputStreamReader(fileIs);
        // read the input stream in Buffered chunks
        BufferedReader br = new BufferedReader(isr);
        // initialize string to put 1 row of file in
        String line = null;
        // set up output stream for writing file to new destination
        InputStream in = new BufferedInputStream(file.getInputStream());
        OutputStream out = new BufferedOutputStream(
                    new FileOutputStream(fileDestination + file.getFilename()));
        int b = 0;
        while ((b = in.read()) != -1)
          out.write(b);
        out.close();
        in.close();
        // loop through file reading 1 row at a time into the String line
        while ((line = br.readLine()) != null) {
          // call to database package in AppModuleImpl.java to insert 1 row into DB
          getAppModule().insertRow(line.toString(),
                                   file.getFilename(),file.getLength(),
                                   file.getContentType());
          System.out.println(line.toString());

          // the commented out text below is for parsing each row,  not needed
          // here, but keeping for an example
        }
        file.dispose();
        br.close();
      } catch (Exception f) {
        f.printStackTrace();
    }
  }
```

means that it listens for a ValueChangeEvent from user interface objects and responds to these changes in the old values. In the fileUploaded method, the file to be uploaded is accessed through the ValueChangeEvent as the new value of the field. The method generates and reads an input stream from the file, copies it character by character into an output stream, and saves the outputstream to a new target file. It also accesses the database connection established elsewhere in the application, and uses the input stream to insert rows of data. While all of this occurs, appropriate data is being associated with variables for the info message that appears in the page after a successful upload. Afterwards, a copy of the uploaded file exists on the server for record-keeping purposes, and the appropriate table in the database has been updated.

Data Access and Retrieval

While it has yet to be implemented, this part of the application, unlike the previous parts, is almost identical to applications already developed and currently in use by this group. The majority of its implementation can be taken from existing applications, leaving more available time and resources for improving the data collection and upload applications.

Discussion

In the particular case of laser safety audits, the Tablet PC was the most appropriate tool, since the forms includes comments. At first glance, however, using the mobile application does not seem to provide significant advantages over the current method. It still consists of two steps, data collection and web upload, and the process in the field takes about as long as it used to, sometimes longer if the user is unfamiliar with the technology. Similarly, a newcomer to the tablet-writing technology may find it difficult to make changes or corrections as instinctively as they used to with pencil and paper. The overall benefits, however, are certain; putting the audit in the database is many times easier and faster, and the data itself much more reliable.

No matter how well designed, the current method of manually entering data into a Filemaker Pro database after first performing the audit by hand is redundant and wasteful. While it's true that doing the form twice allows users more chances to think about their input and correct errors, this can also be done easily with the electronic form, and without redoing the entire form. The upload step itself takes much less time than re-entering all the form data, consisting of only a few mouse clicks. The entired process of database access and data insertion is abstracted away from the user, allowing for effortless upload regardless of the size or complexity of the form. Additionally, the electronic form can be designed to control what can be filled into fields. For example, if there is a known set of buildings where laser safety audits will take place, the Building field can generate a searchable list of values of the possible buildings. This descreases the chance of accidental inaccuracy, as well as eliminates the potential for confusion between equivalent entries like T-2679, 2679, trailer 2679, and so on. Most importantly, all audits performed with this application are now stored in an organized, easily referenced fashion, and can be drawn upon at any time in the future.

As this project continues, there are several action items to complete before this concept can be considered proven and deployable. While they are mostly complete, the data collection forms will be refined for esthetics and the details of data file creation. The web application, on the other hand, has much to improve upon in terms of the business logic, such as validation. With the correct implementation, this application could become a powerful tool for checking

data before committing and providing feedback for the user. Finally, once functional examples are available for both platforms, it is important to sit down with potential users to iron out ergonomics, other desired functionalities, and general opinions of the new technology. Mark Ludwig, head of Laser Safety, and his team of Deputy Laser Safety Officers have expressed interest in the tablet application for their audits, so this application is already being developed with a specific objective in mind. Bi-weekly meetings are currently in place to share progress with Laser Safety, as well as address any changing requirements as development continues. As a support application, this project should keep it's primary goal in mind throughout its development; the improvement of existing, necessary tasks to benefit employees of the lab.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.